An Improved Analysis of the Lossy Difference Aggregator

Hilary Finucane* Michael Mitzenmacher†

ABSTRACT

We provide a detailed analysis of the Lossy Difference Aggregator, a recently developed data structure for measuring latency in a router environment where packet losses can occur. Our analysis provides stronger performance bounds than those given originally, and leads us to a model for how to optimize the parameters for the data structure when the loss rate is not known in advance by using competitive analysis.

1. INTRODUCTION

The Lossy Difference Aggregator (LDA), recently introduced in [2], is a novel hash-based data structure designed to allow measurements of the expectation and standard deviation of latency in a router environment where packet losses occur. While this could be done by recording the time each packet was sent from the sender and the time each packet was received at the receiver and comparing these times packet by packet, such an approach requires too much overhead for this setting. The LDA solution requires significantly less overhead and obtains robust results.

Here we improve on the original analysis given in [2], providing better bounds on the number of samples obtained using the LDA structure when losses occur at a known rate, and giving a framework for optimizing the structure when the loss rate is not known in advance.

We recall the basic setup of the data structure as given in [2].[1] The system is measured at disjoint time intervals; the description below applies to a single time interval. The data is kept in a table of *banks*, where each bank consists of a collection of timestamp accumulator-counter *pairs*; for simplicity, we refer to these timestamp accumulator-counter pairs as buckets when the meaning is clear. The buckets correspond to disjoint samples of packets from the stream; that is, each packet is sampled by at most a single bucket, with some probability that depends on the bucket. The original analysis of [2] assumes that for each packet, first a bank is chosen according to a probability distribution, and then a bucket is chosen uniformly at random from that bank. We can generalize this method by ignoring the banks, instead thinking of each bucket as being chosen with some probability that depends on the bucket (where again the sampling is done so at most one bucket is chosen per packet). Here, we focus on the more general approach. Requiring various collections of buckets to be chosen with the same probability – a requirement which might be useful in hardware – is an optional restriction on our basic layout. In fact, our analysis will show that generally one incurs little loss in performance by utilizing banks, which is indeed fortunate for hardware design.

The sampling is done at the sender and the receiver in a consistent fashion, so that when a packet is not lost the sender and receiver agree on which bucket (if any) is associated with that packet. Each packet has two timestamps recording when it is sent and when it is received, but neither timestamp is sent with the packet, to save space overhead. Instead, for each bucket, the sender keeps track of the sum of the sending times for the packets in that bucket and the total number of packets in the bucket, and the receiver keeps track of the sum of the receiving times for the packets in the bucket and the total number of packets in the bucket. At the end of the time interval, a control packet is sent from the sender that contains the information for each bucket. The receiver can check for each bucket that no packets were lost in the bucket by comparing the number of packets in the bucket. For a bucket where no packets were lost, the difference in the sums divided by the number of packets in the bucket gives the average latency over packets in that bucket. Thus, the sum of all timestamp differences over all buckets that did not lose a packet, divided by the sum of the packet counts over all buckets that did not lose a packet, is an unbiased estimator of the packet latency. (Similar techniques can be used to estimate the variance.)

This approach offers a number of advantages in terms of computation required and space usage. For example, only one bucket is updated for each packet, a single hash function can be used to determine the appropriate bucket, and averages can be taken over all buckets that suffer no loss. The key question is how to choose the sampling probabilities to maximize the number of packets obtained; this is the question we now turn to.

Our results include an improved analysis of the optimal sampling probabilities when the loss rate of packets is known. Based on this analysis, we suggest an optimization framework for the more realistic setting where loss rate of packets is unknown, based on competitive analysis. That is, we aim for an algorithm that, on average, obtains a large fraction of

*School of Engineering and Applied Sciences, Harvard University.

†School of Engineering and Applied Sciences, Harvard University. Supported in part by NSF grant CNS-0721491 and a grant from Cisco Systems. Email: michaelm@eecs.harvard.edu

[1]As mentioned in [2], the approach described there is actually small modification of their original approach suggested by Finucane, and it is this version that we study.

the optimal, where the optimal is average number of packets we would obtain *if we knew the loss rate in advance*. In this framework, we show that for several quite realistic settings, the best competitive ratio is achieved by using a single sampling rate for all of the buckets. This provides theoretical backing for the suggestion of [2] that only a small number of banks, or equivalently only a small number of sampling probabilities, are needed to achieve excellent performance across a range of loss rates.

2. ANALYSIS

2.1 Notation

We will use the following notation, differing slightly from [2]. We let m be the number of buckets, n be the number of packets sent over a given time interval, and L be the number of losses. In cases where we talk of packets undergoing loss at a given rate, we use ℓ to denote the loss rate, and L is a taken to be a binomial random variable with $\mathbf{E}[L] = \ell n$. This is not meant to imply that losses are necessarily independent; they may be correlated in time or in other respects. We do, however, assume that the hashes are uniform and independent in the natural way. That is, if a packet chooses a bucket with probability p (that can depend on the bucket, and may be irrational in our analysis) then we can think of the hash as being uniform on the real interval $[0, 1)$, and the packet is placed in that bucket if it lands in some interval $[a, a + p)$. Again, in this case, the hashes are uniform and independent, and the intervals are disjoint.

In practice, one generally chooses rational probabilities that are computed easily; in [2] the recommendation is to use sampling probabilities with that are integer powers of $1/2$, which can be easily implemented by using the low order bits of a hash of packet. (In fact, more generally one can choose sampling probabilities of the form $i/2^j$ for integers i and j in this fashion.) Here we focus on the ideal of arbitrary sampling probabilities, which give insight and can be rounded or otherwise modified in practice, as we discuss further in the analysis.

In our analysis what is important is the hash value of packet, which determines the bucket the packet will be placed in. It follows that, from the point of view of each bucket, when there is a loss rate of ℓ, it is as though each packet hashed to that bucket is lost with probability ℓ, since each bucket obtains a random sample of the n packets. Our analysis makes use of this fact, allowing us to assume that each packet is lost with probability ℓ.

2.2 The case of small loss

To begin our analysis, we consider the case of a small number of losses. It follows from more general arguments in Sections 2.3 and 2.4 that in this setting we should sample all packets and divide them equiprobably among the buckets; we take that as given for now. When there are exactly L losses, the main concern is the number of buckets which obtain one or more loss. The probability that the L lost packets map to k buckets is given by the Stirling numbers of the second kind; specifically, the Stirling number $S(L, k)$ (also sometimes denoted $\left\{ {L \atop k} \right\}$ [1]) represents the number of ways to partition L items into k non-empty sets. Hence, letting K be a random variable denoting the number of buckets where packets are lost, we have:

$$\mathbf{Pr}(K = k) = \binom{m}{k} k! S(L, k)/m^L.$$

The remaining $n - L$ packets are equally likely to fall into any bucket, and hence the expected number of packets in useful buckets is $(n - L)(1 - \frac{K}{m})$.

When $L = o(\sqrt{m})$, with high probability (that is, with probability $1 - o(1)$) the L losses will occur in distinct buckets, conditioned on which the expected number of packets not in useful bucket is $L + L(n - L)/m$. This matches the statement of [2] that under low loss the expected fraction of packets that do not fall in useful buckets is approximately L/m. More specifically, the number of packets in useful buckets will approximate a binomial random variable $B(n-L, 1-L/m)$. (It will equal this random variable when the L losses occur in different buckets.)

2.3 The case of large known loss, equiprobable buckets

Suppose now we have a constant loss rate ℓ which is not necessarily small, and our goal is to maximize the expected number of packets in useful buckets. What should the sampling rate for each bucket be?

Let us denote our sampling rate for a given bucket by z/n, so that we obtain an average of z samples in the bucket. We will optimize the expectation for a single bucket; standard linearity of expectations arguments give that to maximize the expectation over all buckets we use the same sampling rate for all buckets. (See also our concavity argument in Section 2.4.)

The expected number of useful packets can be found by considering the number of packets that hash to a bucket multiplied by the probability that none are lost. The number of packets in a bucket is given by a binomial distribution $B(n, z/n)$; for convenience we use the Poisson distribution with mean z to approximate this distribution, a standard technique [3], and revisit this approximation later. Let X be the number of useful packets from a given bucket. Then, using the Poisson approximation, for $i > 0$,

$$\mathbf{Pr}(X = i) = \frac{\mathrm{e}^{-z} z^i}{i!}(1 - \ell)^i.$$

That is, for there to be i useful packets, i packets must have been hashed to the bucket, and then with probability $(1-\ell)^i$ none are lost between the source and destination. We then find

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{i=1}^{\infty} i \frac{\mathrm{e}^{-z} z^i}{i!}(1 - \ell)^i \\
&= z(1 - \ell)\mathrm{e}^{-z\ell} \sum_{i=0}^{\infty} \frac{\mathrm{e}^{-z(1-\ell)}(z(1 - \ell))^i}{i!} \\
&= z(1 - \ell)\mathrm{e}^{-z\ell}.
\end{aligned}$$

We can maximize $\mathbf{E}[X]$ by taking derivatives:

$$\frac{d\mathbf{E}[X]}{dz} = (1 - \ell)\mathrm{e}^{-z\ell} - z(1 - \ell)\ell e^{-z\ell},$$

setting this to 0, and canceling. We find the expectation is maximized when $z = 1/\ell$. (We note that there is a requirement that $z \leq n/m$, since we have disjoint buckets; if $\ell > m/n$, we set z to n/m.)

More intuitively, given ℓ, we should choose z so that on average there is exactly one loss per bucket; this gives an average number of useful packets per bucket of $\frac{1-\ell}{\mathrm{e}\ell}$.

We remark that essentially the same argument holds even without the Poisson approximation, and we provide this argument now below. However, we focus on the Poisson approximation argument because, in general, it is simpler, and we believe it should be the starting point for researchers that build on this analysis or use similar analysis for related problems.

Using the binomial distribution instead of the Poisson approximation, we have

$$\mathbf{Pr}(X = i) = \binom{n}{i} \left(\frac{z}{n}\right)^i \left(1 - \frac{z}{n}\right)^{n-i} (1-\ell)^i,$$

and then

$$\begin{aligned} \mathbf{E}[X] &= \sum_{i=1}^{\infty} i \binom{n}{i} \left(\frac{z}{n}\right)^i \left(1 - \frac{z}{n}\right)^{n-i} (1-\ell)^i \\ &= n \sum_{i=1}^{n} \binom{n-1}{i-1} \left(\frac{z(1-\ell)}{n}\right)^i \left(1 - \frac{z}{n}\right)^{n-i} \\ &= z(1-\ell)\left(1 - \ell z/n\right)^{n-1}. \end{aligned}$$

Taking derivatives with respect to z and some algebraic manipulation again gives that $z = 1/\ell$ provides the maximum expectation. We note that $(1 - \ell z/n)^{n-1}$ approaches $e^{-\ell z}$, and although the Poisson approximation slightly overestimates $\mathbf{E}[X]$ for some values of n, ℓ, and z, in practice n is large enough to make the difference negligible. For example, [2] uses $n = 5,000,000$, making the difference between the expected values given by the binomial distribution and the Poisson approximation less than 10^{-7} for all values of ℓ and z. Again, we use the Poisson approximation henceforth for convenience.

Our bound differs slightly from that in [2]. There the authors state a weaker lower bound, which in our notation corresponds to

$$\mathbf{E}[X] \geq \frac{\alpha(1-\alpha)(n-L)}{L+1}$$

for any constant α, $0 < \alpha < 1$. Here the sampling rate per bucket is $\alpha/(L+1)$, and the bound is optimized when $\alpha = 1/2$. We note that the authors don't provide a proof of this bound (in particular, the reason why the authors choose $\alpha/(L+1)$ and not α/L is not clear; it seems the right-hand side of the bound should be the stronger $\frac{\alpha(1-\alpha)(n-L)}{L}$, as we prove below, and we use this bound henceforth), and the bound assumes that the actual number of losses, and not just the loss rate, is known in advance.

Their bound therefore underestimates the number of potential samples in two respects. First, even when using a sampling rate at $\alpha = 1/2$ (or $z = 1/(2\ell)$), we find their bound underestimates the expected number of sampled packets. Assuming that L equals its expectation ℓn, their lower bound on the average is $(1-\ell)/(4\ell)$ per bucket; our analysis yields an expectation of $(1-\ell)\mathrm{e}^{-1/2}/2$, which is a factor of $2\mathrm{e}^{-1/2} \approx 1.21$ higher. (This in fact appears to be the gap between the experimental results and the lower bound of [2] in Figure 4 of [2].) Second, the correct setting in their notation is $\alpha = 1$ (or $z = 1/\ell$), which gives an expected number of packets per bucket of $(1-\ell)/(\mathrm{e}\ell)$, or a factor of $4\mathrm{e}^{-1} \approx 1.47$ higher than the analysis of [2].

It is important to emphasize, however, that their analysis would hold under weaker assumptions than ours. Specifically, their lower bound holds assuming only pairwise independent hash functions and a packet loss process independent of the hashing process, whereas our analysis assumes perfectly random hash functions. While this is not explicitly stated in [2], it follows naturally from their analysis, and we prove it below. While the assumption of perfectly random hash functions appears quite strong, it is often an accurate description of what occurs in practice, even when using weak hash functions such as pairwise independent hash functions. The work of Mitzenmacher and Vadhan [4] provides some theoretical backing for why this is the case, showing that weak random hash functions and semi-random data can combine to yield hash results that are near-uniform.

We now show how pairwise independence suffices for the bound of [2] above. Recall that a hash function H chosen from a uniform pairwise independent family of hash functions with range R satisfies the property that for any pair of items x_1 and x_2 in the domain and any y_1 and y_2 in the range, $\mathbf{Pr}(H(x_1) = y_1 \text{ and } H(x_2) = y_2) = 1/R^2$ (see, e.g., [3]). We assume the choice of whether a packet is sampled and which bucket it is placed in is determined by the result of a pairwise independent hash function applied to the packet. Let L be the (known) number of lost packets, and consider any packet that is not lost. Let packets be sampled at each bucket with probability α/L. Consider the event that an arbitrary one of the $(n-L)$ packets that is not lost is sampled in a useful bucket where no lost packets were sampled at the sender. The probability the packet is sampled by the bucket is α/L; the probability that none of the L lost packets is sampled by this bucket is, by a union bound, at least $1 - L\frac{\alpha}{L} = 1 - \alpha$. Note that this union bound requires only pairwise independence of the hash function H. (See, e.g., [3].) Hence by linearity of expectations the expected number of useful packets X at the bucket satisfies

$$\mathbf{E}[X] \geq \frac{\alpha(1-\alpha)(n-L)}{L}.$$

Finally, we return to the question of performance when the sampling probabilities available are restricted. We consider the specific case where the probabilities are integer powers of $1/2$, although the reasoning can be applied more generally. Our analysis has shown that, with the Poisson approximation, the expected number of useful packets per bucket with sampling probability z/n is $z(1-\ell)\mathrm{e}^{-z\ell}$. Suppose instead of using the optimal $z = 1/\ell$ we use $z = 1/\gamma$, where γ is chosen so that z/n is a power of $1/2$. Then the ratio between this expectation and the optimal expectation is

$$\frac{1-\ell}{\gamma \mathrm{e}^{\ell/\gamma}} \cdot \frac{\mathrm{e}\ell}{1-\ell} = \mathrm{e}\frac{\ell}{\gamma}\mathrm{e}^{-(\ell/\gamma)}.$$

If we choose γ so that z/n is one of the closest powers of $1/2$ to $1/(n\ell)$, then our two choices for ℓ/γ correspond to two possible values x and $2x$, where $1/2 < x \leq 1$. It can be checked that $\min_{1/2<x\leq 1} \max(ex\mathrm{e}^{-x}, 2ex\mathrm{e}^{-2x})$ is achieved when $x = \ln 2$, so that the best choice is not simply to choose z/n to be the nearest power of $1/2$, but to choose z/n to be the next (numerically) smaller power of $1/2$ if that makes $\ln 2 \leq \ell/\gamma \leq 1$, and the next larger power otherwise. The ratio above is then at least $(e \ln 2)/2 \approx 0.942$, and so we reduce our expected number of useful packets per bucket by at

most six percent by restricting to this set of sampling probabilities. This provides some rigorous backing for adopting this computationally simple approach in practice. Indeed, it is similar in spirit to the competitive analysis approach we adopt for the case of unknown loss.

2.4 The case of unknown loss

When the loss rates are unknown ahead of time, as one would expect in practice, there is some choice regarding what one might try to optimize. The original paper [2] suggests some ad hoc heuristics for this setting, but does not provide an optimization framework. Here we suggest various considerations, and argue for an approach based on competitive analysis.

The underlying problem is that it is not possible to simultaneously guarantee optimal behavior for all possible loss rates, so one must decide what to aim for. One natural goal is to maximize the minimum expected number of packets over loss rates in a given range $[\ell_1, \ell_2]$. This goal, however, is straightforward; since the expected number of packets obtained decreases as the loss rate increases regardless of sampling probabilities, it follows that with that goal we simply choose the settings that maximize the expected number of packets for the loss rate ℓ_2, as found in Section 2.3. If instead one had a distribution over possible loss rates in a range $[\ell_1, \ell_2]$, a possible goal might be to maximize the expected number of packets obtained over this distribution. For a collection of m buckets with a sampling rate of z_i/n for the ith bucket this expectation would be given by (using the Poisson approximation)

$$\int_{\ell=\ell_1}^{\ell_2} \sum_{i=1}^{m} z_i(1-\ell)\mathrm{e}^{-z_i\ell} g(\ell)\, d\ell,$$

where $g(\ell)$ would be the density function of the corresponding distribution on ℓ. We may also consider the case as m goes to infinity, in which case the sum can be represented as an integral and we seek to optimize

$$\int_{\ell=\ell_1}^{\ell_2} \int_{z=0}^{1} z(1-\ell)\mathrm{e}^{-z\ell} f(z) g(\ell)\, dz\, d\ell,$$

where $f(z)$ represents the density function of the distribution of the sampling rate over buckets. (Conceivably, this latter expression might be easier to optimize, and then for finite m one could choose sampling rates according to the density function f, directly or in some other fashion.) However, in many settings, we may not have a distribution over loss rates, only a target range for the loss rate, in which case this approach would not apply.

Arguably a more natural and more flexible goal is to minimize the gap between the expected number of packets obtained by the chosen configuration and the optimal configuration, where for the optimal configuration we mean optimal assuming that we knew the loss rate in advance. This corresponds to the well-studied concept of the *competitive ratio* for online algorithms. Suppose that we set a configuration with sampling rate z_i/n for bucket i. Then the ratio between the expected number of packets from useful buckets and the optimal expectation given the loss rate in advance – again using the Poisson approximation – when ℓ is the loss rate is given by:

$$\frac{\sum_{i=1}^{m} z_i(1-\ell)\mathrm{e}^{-z_i\ell}}{m(1/\ell)(1-\ell)\mathrm{e}^{-1}} = \frac{\mathrm{e}}{m}\sum_{i=1}^{m} z_i\ell\mathrm{e}^{-z_i\ell}.$$

We wish this ratio to be as close to 1 as possible. Hence, if we wish to optimize the competitive ratio over a given range $[\ell_1, \ell_2]$, we seek to optimize the competitive ratio C_{m,n,ℓ_1,ℓ_2} over vectors $\vec{z} = (z_1, z_2, \ldots, z_m)$, given by the expression

$$C_{m,n,\ell_1,\ell_2} = \max_{\vec{z}} \min_{\ell\in[\ell_1,\ell_2]} \frac{\mathrm{e}}{m}\sum_{i=1}^{m} z_i\ell\mathrm{e}^{-z_i\ell}.$$

(Technically, our expression C_{m,n,ℓ_1,ℓ_2} is independent of n, since we have used the limiting Poisson approximation; we keep the n in the expression to highlight this.) Let $f(x) = x\mathrm{e}^{-x}$, then this can be restated as finding

$$C = \max_{\vec{z}} \min_{\ell\in[\ell_1,\ell_2]} \frac{\mathrm{e}}{m}\sum_{i=1}^{m} f(z_i\ell),$$

where we use C for C_{m,n,ℓ_1,ℓ_2} where the meaning is clear.

In general, optimizing the competitive ratio C will have to be done numerically. Also, there may be additional constraints. For example, as we have considered previously and as suggested in [2], a natural approach is to use the last few bits of a hash of the packet to determine whether it was sampled by a bucket, in which case we might restrict the sampling probabilities to be an integer power of 1/2. Here we ignore such restrictions and analyze some special cases, which provide useful rules of thumb and some insight. Specifically, we show that as long as the range of values $[\ell_1, \ell_2]$ is sufficiently narrow, the optimal configuration is in fact to use a single sampling probability for all buckets. That is, in the language of [2], there needs to only be a single bank of timestamp accumulator-counter pairs using the same sampling probability.

The mathematics that follows is not meant to obscure the main point, which was suggested as a heuristic in [2]: a small number of sampling probabilities for all the buckets is sufficient to get very good performance. Here, we quantify this heuristic by looking at the competitive ratio with the optimal performance if one knew the loss rate of packets in advance.

2.4.1 The case where $\ell_2/\ell_1 \le 2$

We first show that when the range of loss rates of concern is fairly small, so that $\ell_2 \le 2\ell_1$, then the optimal assignment is for every bucket to have the same sampling probability. This argument therefore extends the setting where every bucket should have the same sampling probability beyond the case of known loss rates covered previously.

Proposition 2.1. *When $\ell_2/\ell_1 \le 2$, then the vector $\vec{z^*} = (Z_1, Z_2, \ldots, Z_m)$ that maximizes the competitive ratio has $Z_i = \frac{\ln\ell_2 - \ln\ell_1}{\ell_2 - \ell_1}$ for all i.*

PROOF. We first note that f is increasing for $0 < x < 1$, decreasing for $x > 1$, and concave for $0 \le x \le 2$. Any assignment of z that is optimal must have $z_i\ell_1 \le 1$ for all i, because otherwise the value of $\sum_{i=1}^{m} f(z_i\ell)$ can be increased for all $\ell \in [\ell_1, \ell_2]$ by decreasing the value of this z_i. Since we have assumed that $\ell_2/\ell_1 \le 2$, this implies that $z_i\ell \le 2$ for all i and all $\ell \in [\ell_1, \ell_2]$, which it turn implies concavity of $\sum_{i=1}^{m} f(z_i\ell)$ with respect to ℓ at the optimal assignment of the z_i. Hence the minimum over $\ell \in [\ell_1, \ell_2]$ at the optimal $\vec{z}$, or for any $\vec{z}$ where $z_i\ell \le 2$ for all i, must be obtained at ℓ_2 or ℓ_1.

Let $w = \frac{\ln\ell_2 - \ln\ell_1}{\ell_2 - \ell_1}$; note that $w\ell_1 < 1$, $w\ell_2 > 1$, and $f(w\ell_1) = f(w\ell_2)$. Suppose $\vec{z^*} = (Z_1, Z_2, \ldots, Z_m)$ is the

optimal vector $\vec{z}$ for the competitive ratio.

Suppose that the minimum over ℓ at $\vec{z^*}$ is achieved at ℓ_1, and $\sum_i f(Z_i\ell_1) > mf(w\ell_1)$. Then

$$\begin{aligned} mf(w\ell_1) &< \sum_i f(Z_i\ell_1) \\ &\leq mf\left(\frac{1}{m}\sum_i Z_i\ell_1\right) \end{aligned}$$

where the second inequality follows from concavity.

Since $w\ell_1 < 1$ and f is strictly increasing on $[0,1]$, this implies that $1/m\sum_i Z_i > w$, which in turn implies that $f(1/m\sum_i Z_i\ell_2) < f(w\ell_2)$ since $w\ell_2 > 1$ and f is strictly decreasing on $[1,\infty)$. Again applying concavity, we obtain

$$\begin{aligned} \sum_i f(Z_i\ell_2) &\leq mf\left(\frac{1}{m}\sum_i Z_i\ell_2\right) \\ &< mf(w\ell_2) \\ &= mf(w\ell_1) \\ &< \sum_i f(Z_i\ell_1) \end{aligned}$$

But this contradicts our assumption that the minimum was achieved at ℓ_1. An identical proof shows that a minimum greater than $mf(w\ell_1)$ cannot be achieved at ℓ_2, so the optimal minimum is achieved when $Z_i = w$ for all i.

To gain some insight, suppose $\ell_2 = r\ell_1$ for $r > 1$. Then $w\ell_1 = (\ln r)/(r-1)$ and the competitive ratio in this case is therefore given by

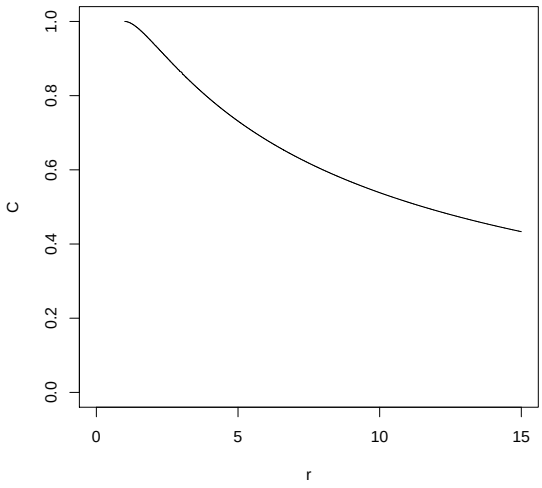
$$\begin{aligned} C &= \frac{\mathrm{e}}{m}\sum_{i=1}^{m} f(w\ell_1) \\ &= \mathrm{e}f(w\ell_1) \\ &= \frac{\mathrm{e}\ln r}{(r-1)r^{1/(r-1)}}. \end{aligned}$$

(See Figure 1.) For $r = 2$, this gives a competitive ratio of $(e\ln 2)/2 \approx 0.942$. (This is the same value as in Section 2.2; indeed, this can be viewed as a generalization of the result there.) Even for $r = 10$, where the use of a single bank is not guaranteed to be optimal by Proposition 2.1 and the range of interest of loss rates spans an order of magnitude, the competitive ratio is greater than $0.538 > 1/2$. Notice also that as r goes to 1, the competitive ratio converges to 1; that is, with a fixed loss rate, we obtain the optimal solution where all buckets have the same loss rate.

We emphasize that as the size of the range of loss rates increases, it will no longer be the case that having all buckets have the same sampling rate will prove optimal. For example, even in the case of two buckets, if $\ell_1 = 0.01$ and $\ell_2 = 0.4$, then $z_1 = z_2 = w \approx 9.459$ gives a competitive ratio of only 0.234, while choosing $z_1 = 4.019$ and $z_2 = 44.032$ gives a competitive ratio of over 0.437. (The values for z_1 and z_2 were found with an optimization routine.) However, for smaller ranges of loss probabilities, using a bucket sampling probability near w/n for all buckets appears quite effective.

2.4.2 The case where $m = 2, \ell_2/\ell_1 < 5.5$

A further interesting case is when we allow only two different types of banks, each corresponding to a sampling rate, with the same number of buckets in each bank. This setting



.

Figure 1: The competitive ratio plotted as a function of the ratio $r = \ell_2/\ell_1$.

was suggested as an ad hoc heuristic in [2]. Interestingly, we find here that even for larger ranges of loss rates, the optimal configuration in terms of the competitive ratio requires using only a single sampling probability. It suffices to consider the case of exactly two buckets. We show this result when $\ell_2/\ell_1 < 5.5$; we have not attempted to optimize the constant 5.5, but as our previous example shows, for large enough ranges the result will not hold. Throughout this section, we assume $z_1 \geq z_2$.

Proposition 2.2. *When $\ell_2/\ell_1 < 5.5$ and $m = 2$, then the vector (Z_1, Z_2) that maximizes the competitive ratio has $Z_1 = Z_2 = \frac{\ln\ell_2 - \ln\ell_1}{\ell_2 - \ell_1}$.*

Proof. Let $r = \ell_2/\ell_1$. By letting $z_1 = z_2 = \frac{\ln(\ell_2)-\ln(\ell_1)}{\ell_2-\ell_1}$, we can guarantee a competitive ratio of $\frac{\mathrm{e}f(\ell z_1)+f(\ell z_2))}{2} > \frac{\mathrm{e}\ln(r)}{(r-1)r^{1/(r-1)}} = h(r)$ for all $\ell \in [\ell_1, \ell_2]$ (as shown in Section 2.4.1). We can check that h is monotonically decreasing and $h(5.5) > 0.7$ (see Figure 1), so since $r < 5.5$, we can always guarantee a minimum competitive ratio of at least 0.7.

We can also check that when $z_1 = 7$ and $z_2 = 1$, the maximum of $\frac{\mathrm{e}(f(\ell z_1)+f(\ell z_2))}{2}$ over all $\ell \geq 0$ is less than 0.7, and that this maximum value decreases as z_1 increases. By scaling accordingly with ℓ, this shows us that for any z_1 and z_2 such that $z_1/z_2 > 7$, we cannot achieve a competitive ratio better than 0.7. Since we know we can choose z_1 and z_2 to guarantee better than 0.7 by setting $z_1 = z_2 = \frac{\ln(\ell_2)-\ln(\ell_1)}{\ell_2-\ell_1}$, we can discard all pairs of z_i such that $z_1/z_2 > 7$.

Note also that for fixed z_1, z_2 such that $z_1/z_2 \leq 7$, $f(\ell z_1) + f(\ell z_2)$ has no local minima over ℓ. Since we have discarded all but these pairs, we can now ignore any such potential local minima and assume that the minimum over ℓ at (Z_1, Z_2) is achieved at ℓ_2 or ℓ_1.

Suppose that the minimum over ℓ at (Z_1, Z_2) is achieved at ℓ_2, and that $f(Z_1\ell_1) + f(Z_2\ell_1)$ is strictly greater than $f(Z_1\ell_2) + f(Z_2\ell_2)$. We must have $Z_1\ell_2 > 1$, or otherwise we

could increase Z_1 to increase the value of both $f(Z_1\ell_1)$ and $f(Z_1\ell_2)$, so $f(Z_1\ell_2)+f(Z_2\ell_2)$ can be increased by decreasing Z_1 slightly. By the continuity of f, we can make the size of the change small enough that $f(Z_1\ell_1) + f(Z_2\ell_1)$ is still greater than $f(Z_1\ell_2) + f(Z_2\ell_2)$, so the minimum over $\ell \in [\ell_1, \ell_2]$ is still achieved at ℓ_2. But then by decreasing Z_1, we have improved the minimum over ℓ, so (Z_1, Z_2) must not have been the optimal assignment. A similar argument shows that $f(Z_1\ell_1)+f(Z_2\ell_1) \not< f(Z_1\ell_2)+f(Z_2\ell_2)$, and thus our optimal pair (Z_1, Z_2) must satisfy $f(Z_1\ell_1) + f(Z_2\ell_1) = f(Z_1\ell_2) + f(Z_2\ell_2)$.

Let us consider z_2 as a function of z_1, so that $z_2(z_1)$ satisfies this equality. We want to find z_1 that maximizes $f(z_1\ell_2) + f(z_2(z_1)\ell_2)$. This gives us two equations:

$$\begin{aligned} f(z_1\ell_1) + f(z_2(z_1)\ell_1) &= f(z_1\ell_2) + f(z_2(z_1)\ell_2); \\ \frac{d}{dz_1}\left(f(z_1\ell_1) + f(z_2(z_1)\ell_1)\right) &= 0. \end{aligned}$$

With implicit differentiation of the first equation, we get

$$\ell_1 f'(z_1\ell_1)+\ell_1 z_2'(z_1) f'(z_2(z_1)\ell_1) = \ell_2 f'(z_1\ell_2)+\ell_2 z_2'(z_1) f'(z_2(z_1)\ell_2)$$

where $f'(x) = (1-x)\mathrm{e}^{-x}$. So

$$\begin{aligned} z_2'(z_1) &= \frac{\ell_2 f'(z_1\ell_2) - \ell_1 f'(z_1\ell_1)}{\ell_1 f'(z_2(z_1)\ell_1) - \ell_2 f'(z_2(z_1)\ell_2)} \\ &= \frac{\ell_2(1-z_1\ell_2)\mathrm{e}^{-z_1\ell_2} - \ell_1(1-z_1\ell_1)\mathrm{e}^{-z_1\ell_1}}{\ell_1(1-z_2(z_1)\ell_1)\mathrm{e}^{-z_2(z_1)\ell_1} - \ell_2(1-z_2(z_1)\ell_2)\mathrm{e}^{-z_2(z_1)\ell_2}}. \end{aligned}$$

Now substituting this in the second equation, we get the following chain:

$$\begin{aligned} \ell_1 f'(z_1\ell_1) + \ell_1 z_2'(z_1) f'(z_2(z_1)\ell_1) &= 0; \\ f'(z_1\ell_1) + z_2'(z_1) f(z_2(z_1)\ell_1) &= 0; \\ f'(z_1\ell_1) + \frac{\ell_2 f'(z_1\ell_2) - \ell_1 f'(z_1\ell_1)}{\ell_1 f'(z_2\ell_1) - \ell_2 f'(z_2\ell_2)} f'(z_2\ell_1) &= 0; \\ f'(z_1\ell_1)(\ell_1 f'(z_2\ell_1) - \ell_2 f'(z_2\ell_2)) & \\ + f'(z_2\ell_1)(\ell_2 f'(z_1\ell_2) - \ell_1 f'(z_1\ell_1)) &= 0. \end{aligned}$$

Further algebraic manipulation yields

$$\begin{aligned} f'(z_1\ell_1)(\ell_2 f'(z_2\ell_2) - \ell_1 f'(z_2\ell_1)) &= f'(z_2\ell_1)(\ell_2 f'(z_1\ell_2) - \ell_1 f'(z_1\ell_1)); \\ \ell_2 f'(z_2\ell_2)/f'(z_2\ell_1) - \ell_1 &= \ell_2 f'(z_1\ell_2)/f'(z_1\ell_1) - \ell_1; \\ f'(z_2\ell_2)/f'(z_2\ell_1) &= f'(z_1\ell_2)/f'(z_1\ell_1); \\ \mathrm{e}^{-z_2(\ell_2-\ell_1)}\frac{1-z_2\ell_2}{1-z_2\ell_1} &= \mathrm{e}^{-z_1(\ell_2-\ell_1)}\frac{1-z_1\ell_2}{1-z_1\ell_1}. \end{aligned}$$

Let

$$g(x) = \mathrm{e}^{-x(\ell_2-\ell_1)}\frac{1-x\ell_2}{1-x\ell_1},$$

and note that from the above we have $g(z_2) = g(z_1)$. We will show that g' is never 0 for given our assumption that $\ell_2/\ell_1 < 5.5$, which by the Mean Value Theorem implies that $z_2 = z_1$. Suppose $g'(x) = 0$. Then

$$\begin{aligned} -(\ell_2-\ell_1)\mathrm{e}^{-x(\ell_2-\ell_1)}\frac{1-x\ell_2}{1-x\ell_1} & \\ + \mathrm{e}^{-x(\ell_2-\ell_1)}\frac{-\ell_2(1-x\ell_1)+\ell_1(1-x\ell_2)}{(1-x\ell_1)^2} &= 0. \end{aligned}$$

Standard algebraic manipulation simplifies the above to

$$x^2\ell_2\ell_1 - x(\ell_2+\ell_1) + 2 = 0.$$

This equation has real roots only if

$$\ell_2^2 - 6\ell_2\ell_1 + \ell_1^2 \geq 0.$$

Now let $s = \ell_2/\ell_1$. The above equation is equivalent to

$$s^2\ell_1^2 - 6s\ell_1^2 + \ell_1^2 \geq 0,$$

or

$$s^2 - 6s + 1 \geq 0.$$

Solving for s, we see this requires that either $s < 1$, or $s > \frac{6+\sqrt{32}}{2} > 5.8$. As we have limited s to be between 1 and 5.5, we have $g'(x) \neq 0$, so at our optimal pair (Z_1, Z_2) we must have $Z_1 = Z_2 = y$ for some y. Since $f(Z_1\ell_1) + f(Z_2\ell_1) = f(Z_1\ell_2) + f(Z_2\ell_2)$, we know y must satisfy

$$2y\ell_1 e^{-y\ell_1} = 2y\ell_2 e^{-y\ell_2}$$

Solving this for y, we get $y = \frac{\log\ell_1 - \log\ell_2}{\ell_1 - \ell_2}$.

3. CONCLUSION

We have considered a recently proposed data structure, the Lossy Difference Aggregator, and provided a more complete analysis of it. In particular, we have shown how an improved optimization analysis in the case of a known loss rate can improve the expected number of packets obtained substantially, and we have introduced a competitive analysis framework that can allow for robust performance across a range of possible loss rates. We have also shown that, in some natural cases, our framework suggests that one type of bank, with one sampling probability, is sufficient for the optimal competitive ratio. We believe the Lossy Difference Aggregator may find many other applications beyond the network router setting, and if so, we hope this improved analysis will allow better use of the data structure in these applications.

4. ACKNOWLEDGMENTS

Thanks to Sharon Goldberg, Kirill Levchenko, and George Varghese for comments on earlier drafts of this work.

5. REFERENCES

[1] D. E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd Edition)*. Addison-Wesley, 1997.

[2] R. Kompella, K. Levchenko, A. Snoeren, and G. Varghese. Every Microsecond Counts: Tracking Fine-Grain Latencies with a Lossy Difference Aggregator. In *Proceedings of SIGCOMM 2009*.

[3] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[4] M. Mitzenmacher and S. Vadhan. Why simple hash functions work: exploiting the entropy in a data stream. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 746–755, 2008.